REMARKS

Reconsideration and allowance of the above-referenced application are respectfully requested.

Upon entry of this amendment, claims 1,2, 4-11, 13-20, and 22-39 will remain in the application.

Claim Objections

Claim 30 was objected to for informalities. Claim 30 has been corrected.

Response to Applicant's Remark

The Action indicated the Applicant's remark on page 17 of the previous Response re the status of the claims was in error for not indicating that claims 2, 12, and 21 were canceled.

The indication of "Claims canceled: None" the Applicant's remark was a reference to the current action taken by the Response. The "Claims previously in issue" and "Claims remaining in issue" items in the same table in the Response make the status of the claims clear. Furthermore, claim 3, not 2, is canceled. Accordingly, Applicant submits that no correction is necessary.

Claim Rejections – 35 U.S.C. § 103

Claims 1, 2, 4, 8-11, 13, 17-20, 22, 26-29, 32, 33, 36, and 37 were rejected under 35 U.S.C. §103(a) as allegedly being obvious in view of Ruf (U.S. Patent No. 6,077,313) in view of Bodin et al. ("A User Level Program Transformation Tool", hereinafter "Bodin"). Applicant respectfully traverses this rejection with respect to the claims in issue.

Neither reference, alone or in combination, teaches or suggests the invention as claimed. Ruf is directed to a method for analyzing the source code of a program in a particular way ("type" dataflow analysis) in order to "better optimize, understand, or browse" the computer program (Ruf, col. 1, ll. 14-15). Ruf's specific technique is directed to partitioning data types into phases so that dataflow analysis may be performed independently on each such phase (Ruf,

col. 6, ll. 61-67), thus reducing the "cost" of analysis in terms of memory usage and execution time of the analysis (Ruf, col. 1, ll. 51-53, and col. 2, ll. 3-7; col. 4, ll. 13-19). Such partitioning allows analysis of the independent phases to be conducted in parallel (Ruf, col. 7, ll. 6-13) – which is the only type of parallelism discussed by Ruf.

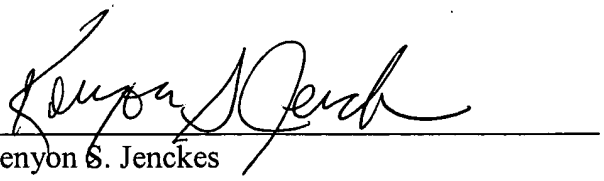
Importantly, Ruf neither teaches or suggests that the output of his process is a parallel dataflow graph that may be executed by a parallel runtime system. Ruf teaches that a program can be analyzed in parallel by performing type dataflow analysis (for example, by a compiler in order to optimize the serial execution of the program) on multiple independent phases concurrently. However, Ruf does not teach or suggest how to produce a parallel computation specification based on such analysis as part of transforming a serial dataflow graph into a parallel data flow graph.

Bodin is not even pertinent art. Bodin is directed at a script language that allows a programmer to use pattern matching to find text statements in large Fortran programs (or large numbers of such programs) in order to insert other text. Examples include inserting calls to library routines, inserting "instrumentation" code (e.g., code that re-cords events like how many times a loop is executed), and restructuring loops (Bodin p. 185, sec. 4 et seq.). Nothing in Bodin teaches or suggests parallelizing a computer application program based on a script of a script-driven software tool, comprising automatically analyzing the script and producing a parallel computation specification based on such analysis.

Neither Ruf nor Bodin, either alone or in combinations, teaches or suggests constructing a parallel dataflow graph that may be executed by a parallel runtime system. Accordingly, Applicant submits that claims 1, 2, 4, 8-11, 13, 17-20, 22, 26-29, 32, 33, 36, and 37 are allowable.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,



Date: September 3, 2004

Kenyon S. Jenckes
Reg. No. 41,873

PTO Customer No. 20985

Fish & Richardson P.C.
12390 El Camino Real
San Diego, California 92130
Telephone: (858) 678-5070
Facsimile: (858) 678-5099

10432341.doc